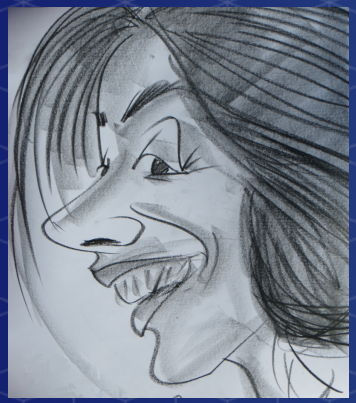
Pique curiosity, not diabetic fingers

Axelle Apvrille (Fortinet)
Travis Goodspeed

July 2020

Hello!



Axelle Apvrille
Principal Security Researcher at
Fortinet, @cryptax
Mobile malware, IoT, Ph0wn CTF

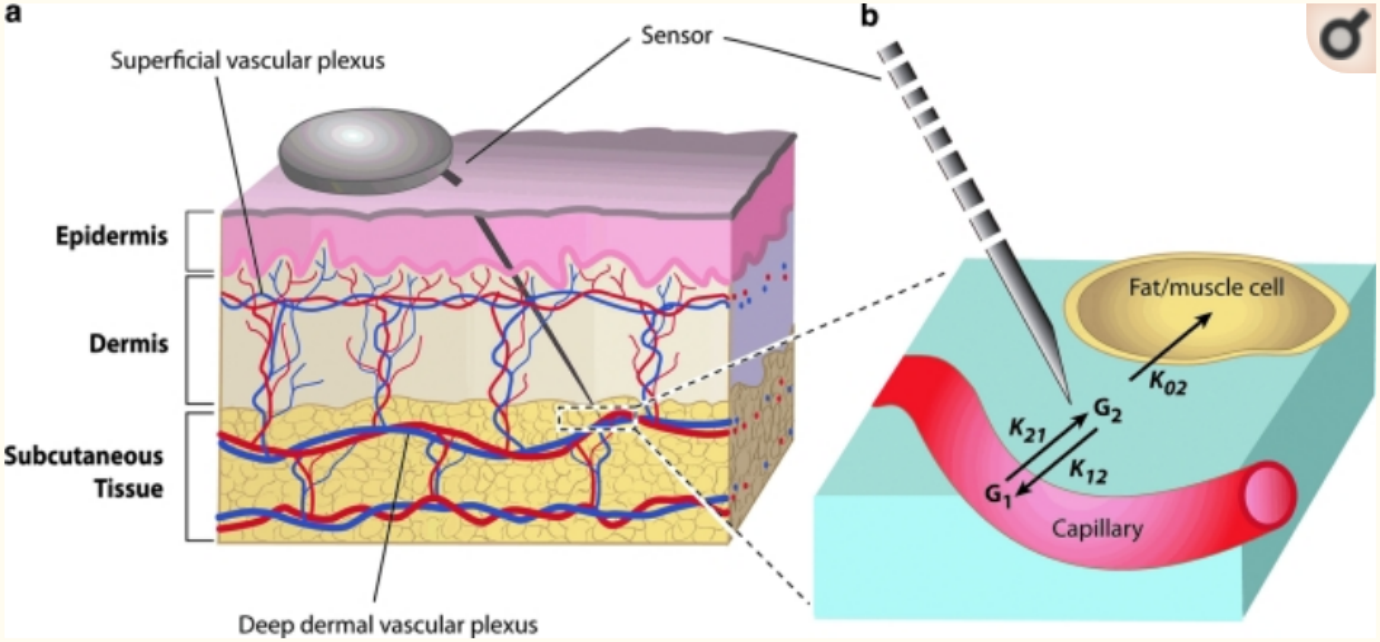
Travis Goodspeed
Digital watchmaker and Studebaker
enthusiast, @travisgoodspeed
GoodFET, GoodWatch, PoC||GTFO

Flash Glucose Monitoring systems

Diabetes Technol Ther. 2009 Jun; 11(Suppl 1): S-11–S-16.
doi: 10.1089/dia.2009.0002
PMCID: PMC2903977
PMID: 19469670

A Tale of Two Compartments: Interstitial Versus Blood Glucose Monitoring

Eda Cengiz, M.D. and William V. Tamborlane, M.D.



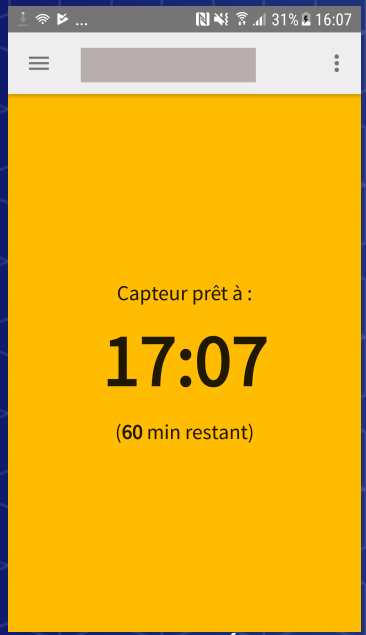
@cryptax testing the sensor!

Screenshot from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2903977/

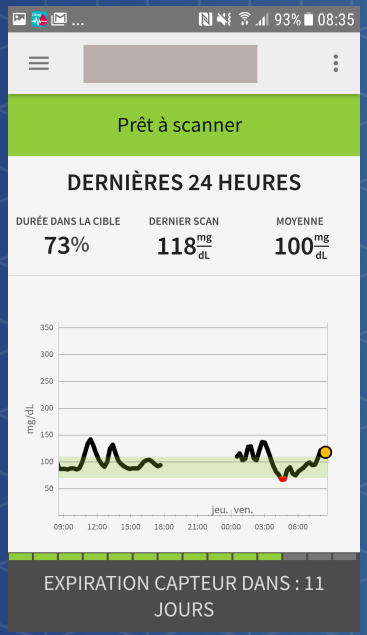
Sensor life cycle

Assemble pack

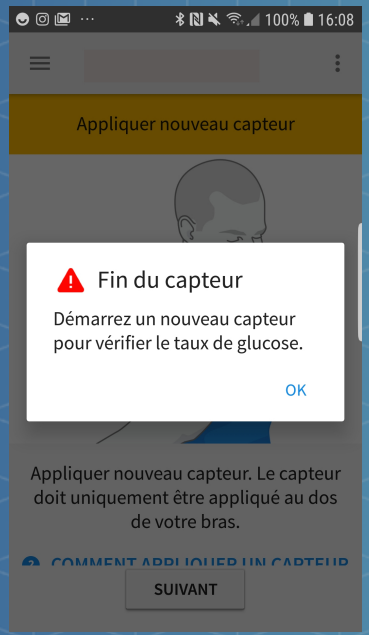
Apply sensor



Activate it (60 min)



Use it



Expires after 14 days

Wanna hack? Working around limitations

1. Max life time
2. Warm up time
3. Geographical location

Disclaimer

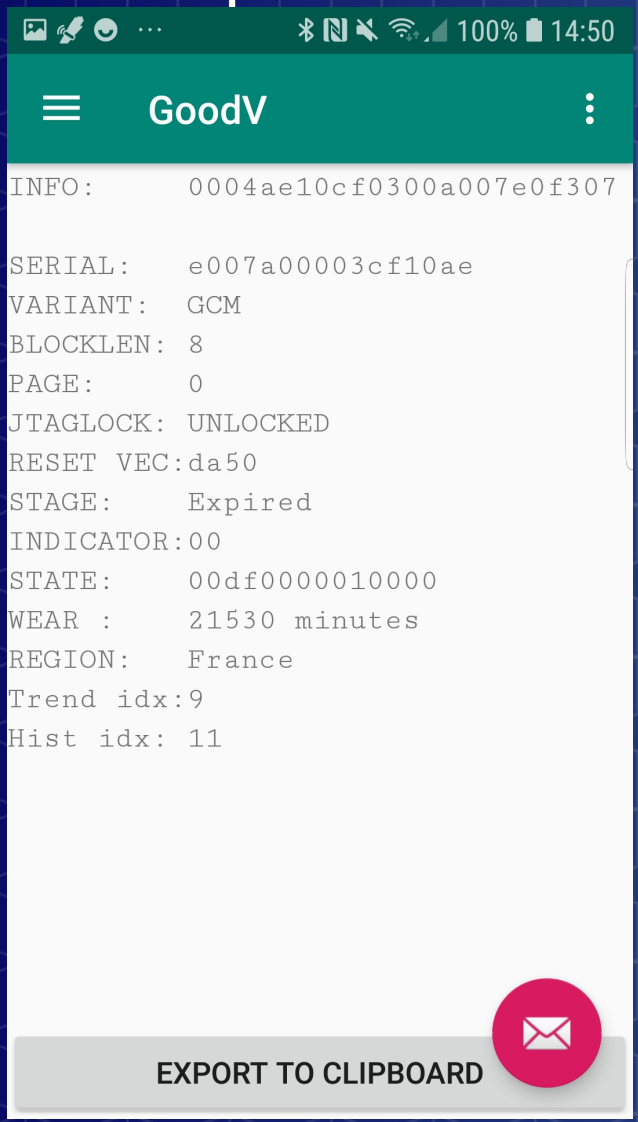
Those hacks work on the **technical** side

They haven't been tested from a medical point of view, and we strongly discourage diabetic users to play with them

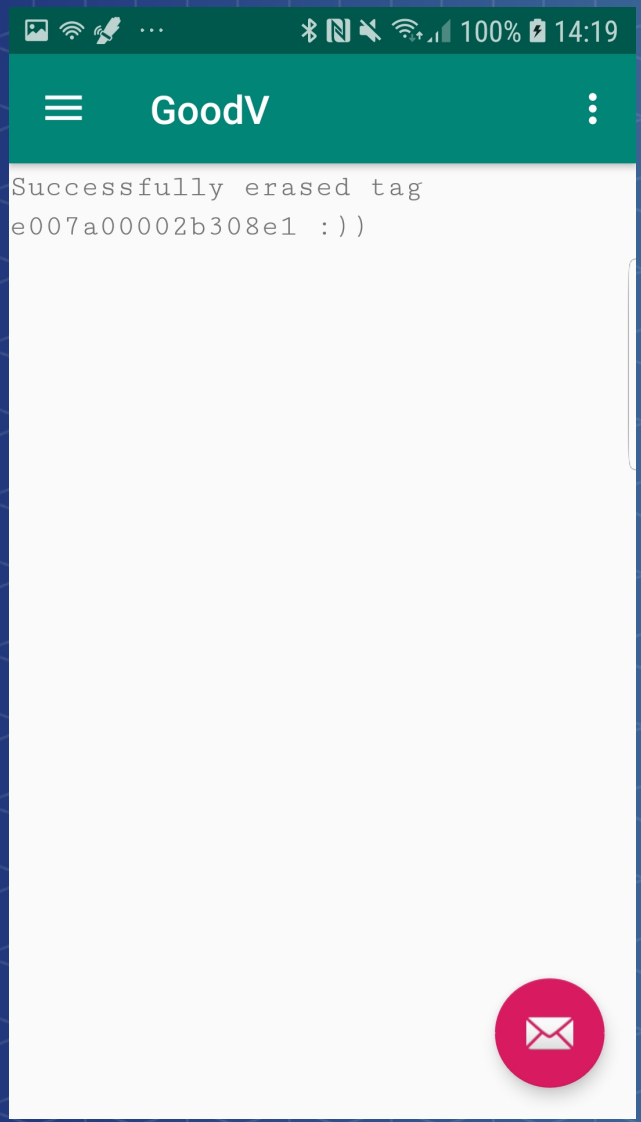
but an attacker could...

Resurrection Demo

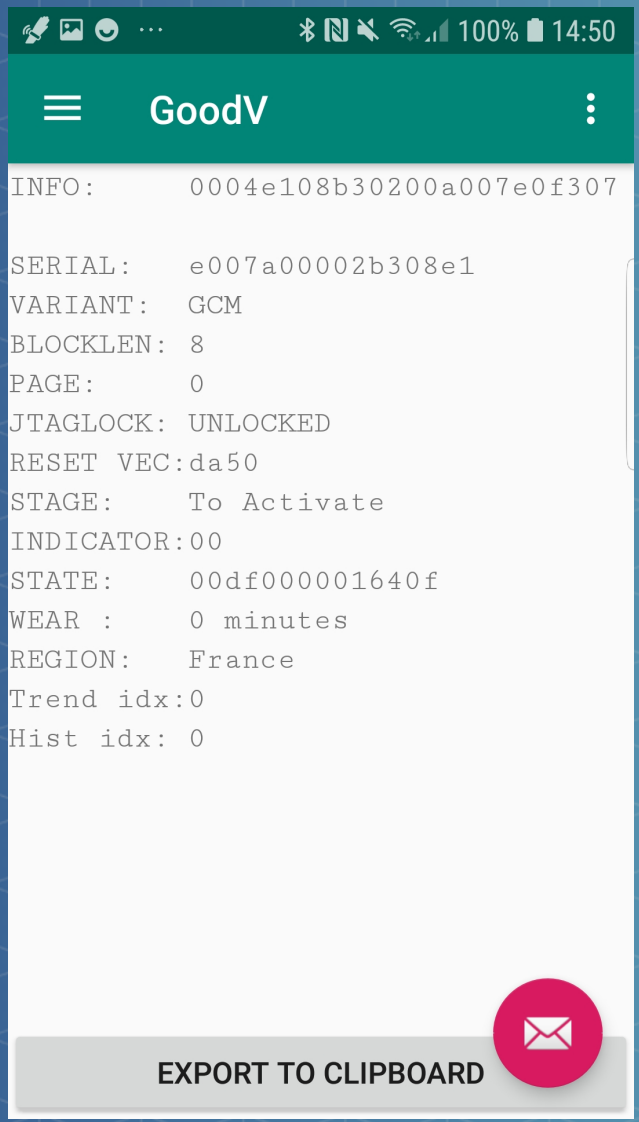
Backup slides :P



Expired



Reset the sensor



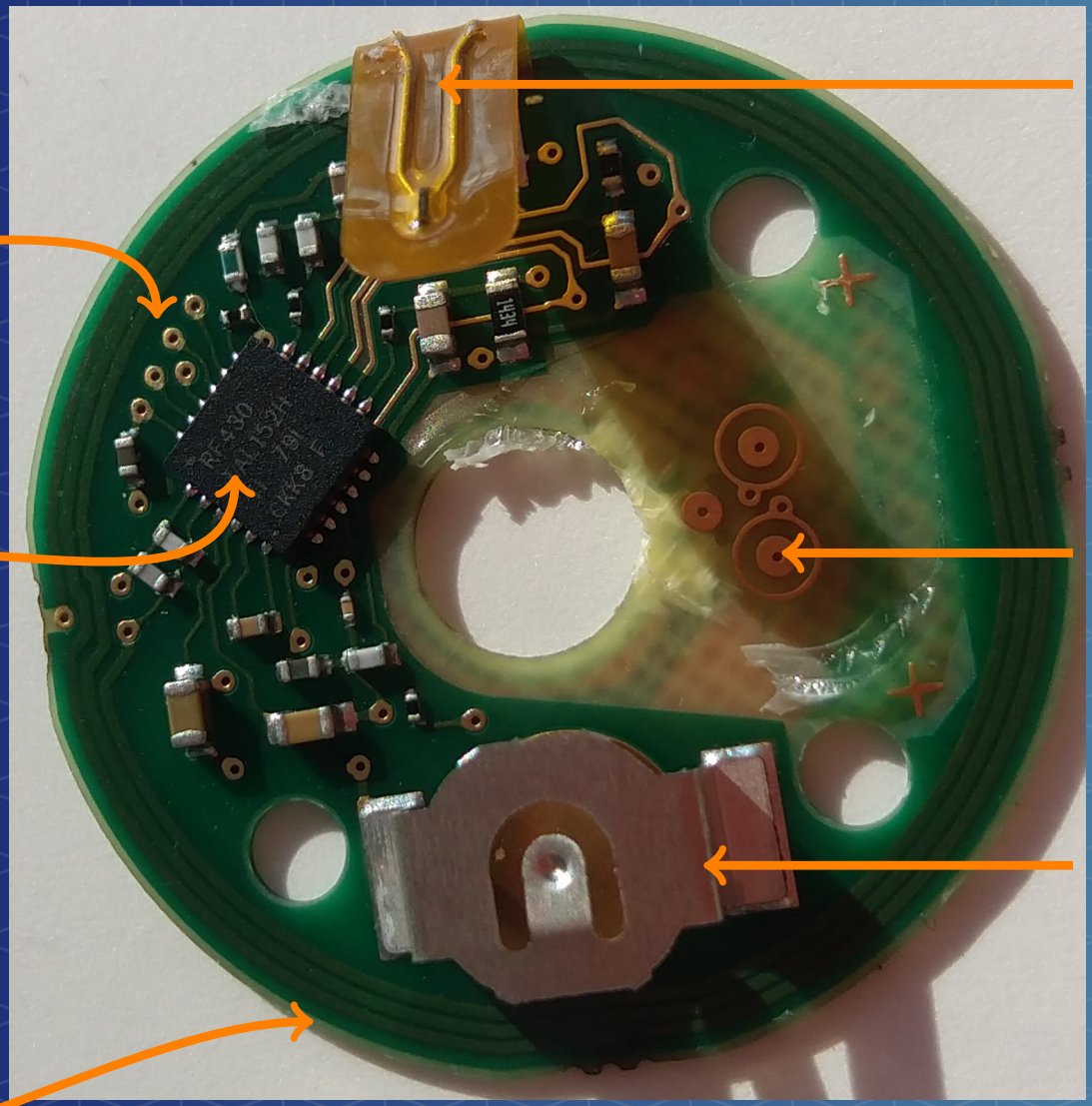
"To Activate" stage now

How does that work?

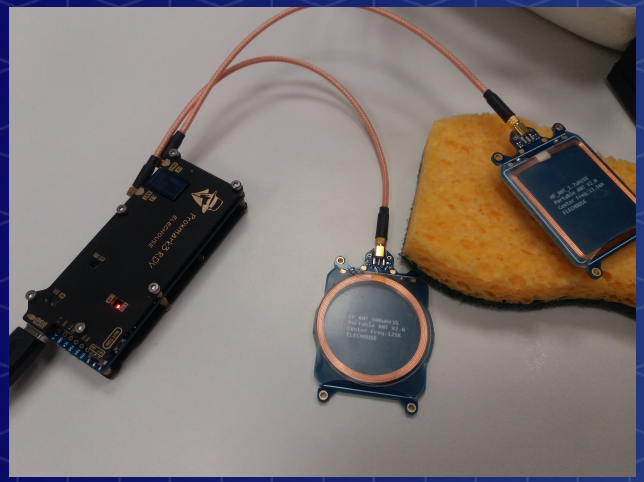
Let's speed through previous work

More information: watch our talk at BlackAlps 2019

Tear down the sensor



Blocks exposed by NFC

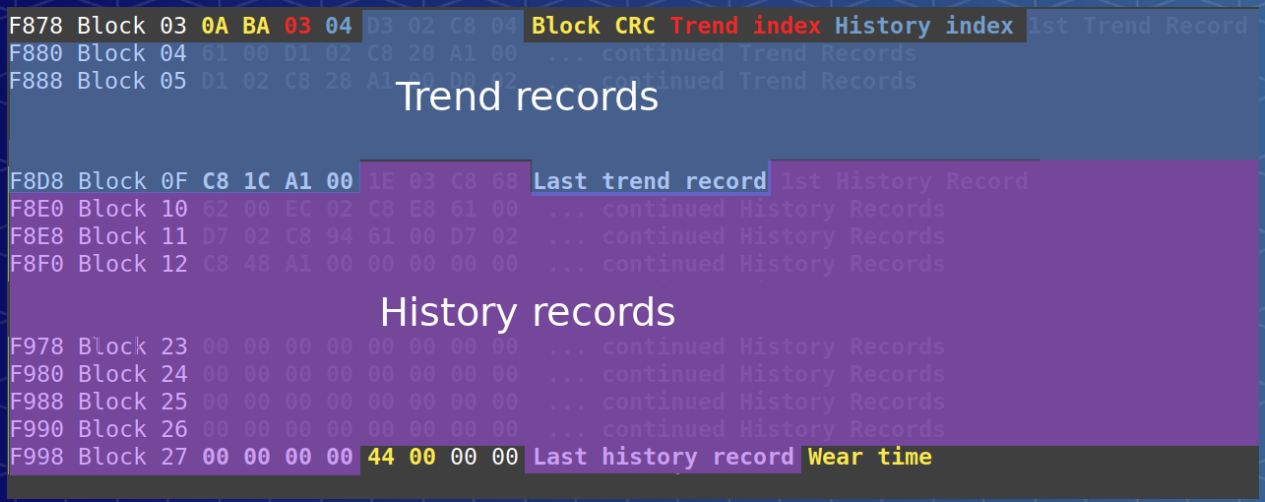


Sponge wet with hot sugar to simulate glucose

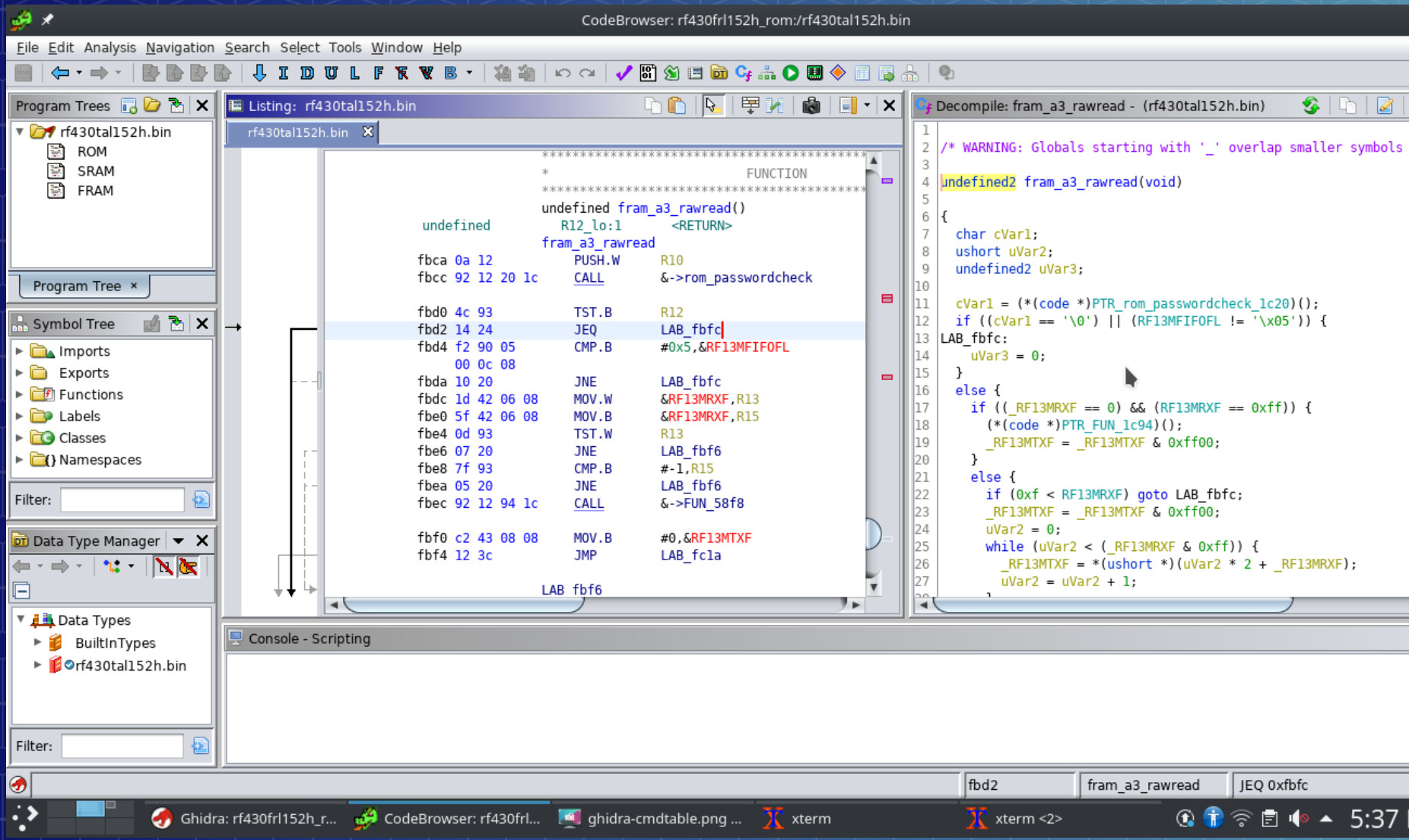
```
Tag UID : E007A00003183AD2
Tag Info: Texas Instrument France

Valid ISO15693 Tag Found - Quiting Search

Reading memory from tag UID=E007A00003183AD2
Tag Info: Texas Instrument France
Block 00   F4 18 B0 32 03 01 02 08     ...2....
Block 01   00 00 00 00 00 00 00 00     ........
Block 02   00 00 00 00 00 00 00 00     ........
Block 03   F9 2B 0E 08 1F 00 C0 96     .+......
Block 04   AB 80 1E 00 C0 92 AB 80     ........
Block 05   1F 00 C0 96 AB 80 1F 00     ........
Block 06   C0 92 AB 80 1E 00 C0 8E     ........
```

Working out memory layout

Section	Begin	End
Activation blocks	F860	F877
Glucose records	F878	F99F
Sensor region	F9A0	F9B7
Commands	F9B8	FFCF
Footer	FFD0	FFF7



A3 Raw Read


CodeBrowser: rf430frl152h_rom:/rf430tal152h.bin
File Edit Analysis Navigation Search Select Tools Window Help
Program Trees
rf430tal152h.bin
ROM
SRAM
FRAM
Program Tree
Symbol Tree
Imports
Exports
Functions
Labels
Classes
Namespaces
Filter:
Data Type Manager
Data Types
BuiltInTypes
rf430tal152h.bin
Filter:
Listing: rf430tal152h.bin
rf430tal152h.bin
FUNCTION
undefined fram_a3_rawread()
undefined R12_lo:1 <RETURN>
fram_a3_rawread
fbca 0a 12 PUSH.W R10
fbcc 92 12 20 1c CALL &->rom_passwordcheck
fbd0 4c 93 TST.B R12
fbd2 14 24 JEQ LAB_fbfc
fbd4 f2 90 05 00 0c 08 CMP.B #0x5,&RF13MFIFOFL
fbda 10 20 JNE LAB_fbfc
fbdc 1d 42 06 08 MOV.W &RF13MRXF,R13
fbe0 5f 42 06 08 MOV.B &RF13MRXF,R15
fbe4 0d 93 TST.W R13
fbe6 07 20 JNE LAB_fbf6
fbe8 7f 93 CMP.B #-1,R15
fbea 05 20 JNE LAB_fbf6
fbec 92 12 94 1c CALL &->FUN_58f8
fbf0 c2 43 08 08 MOV.B #0,&RF13MTXF
fbf4 12 3c JMP LAB_fc1a
LAB fbf6
Decompile: fram_a3_rawread - (rf430tal152h.bin)
/* WARNING: Globals starting with '_' overlap smaller symbols
undefined2 fram_a3_rawread(void)
{
char cVar1;
ushort uVar2;
undefined2 uVar3;
cVar1 = (*(code *)PTR_rom_passwordcheck_1c20)();
if ((cVar1 == '\0') || (RF13MFIFOFL != '\x05')) {
LAB_fbfc:
uVar3 = 0;
}
else {
if ((_RF13MRXF == 0) && (RF13MRXF == 0xff)) {
(*(code *)PTR_FUN_1c94)();
_RF13MTXF = _RF13MTXF & 0xff00;
}
else {
if (0xf < RF13MRXF) goto LAB_fbfc;
_RF13MTXF = _RF13MTXF & 0xff00;
uVar2 = 0;
while (uVar2 < (_RF13MRXF & 0xff)) {
_RF13MTXF = *(ushort *)(uVar2 * 2 + _RF13MRXF);
uVar2 = uVar2 + 1;
Console - Scripting
fbd2
fram_a3_rawread
JEQ 0xfbfc
Ghidra: rf430frl152h_r...
CodeBrowser: rf430frl...
ghidra-cmdtable.png ...
xterm
xterm <2>
5:37

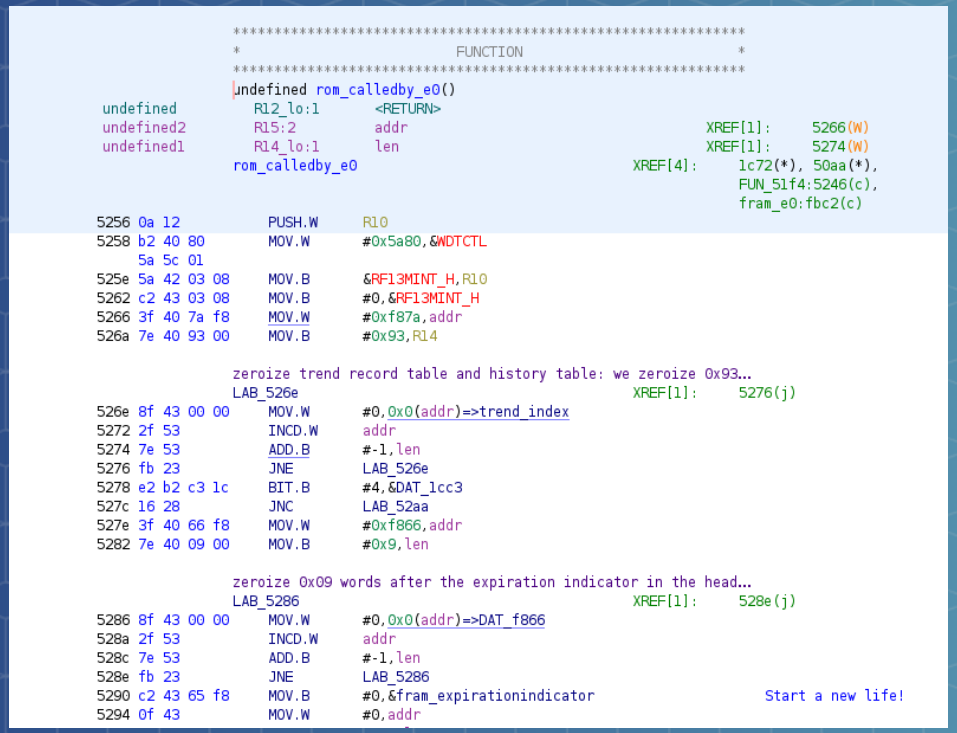
Dump firmware

You're up to level!

Now, let's have a close look to E0

E0 command

- E0 is disabled, but the code is included in the firmware
- It *resets* the sensor
- Disassembly in tech report
- Activity blocks have two important bytes:
  1. Stage of Life. 1 to activate, 3 operational, 5 expired...
  2. Activity switch. 0 inactive, 1 active
- Each section is protected by a CRC



```
                        *************************************************************
                        *                          FUNCTION                         *
                        *************************************************************
                        undefined rom_calledby_e0()
undefined                 R12_lo:1        <RETURN>
undefined2                R15:2           addr                          XREF[1]:     5266(W)
undefined1                R14_lo:1        len                           XREF[1]:     5274(W)
                        rom_calledby_e0                               XREF[4]:     1c72(*), 50aa(*),
                                                                                   FUN_51f4:5246(c),
                                                                                   fram_e0:fbc2(c)
5256 0a 12              PUSH.W      R10
5258 b2 40 80           MOV.W       #0x5a80,&WDTCTL
     5a 5c 01
525e 5a 42 03 08        MOV.B       &RF13MINT_H,R10
5262 c2 43 03 08        MOV.B       #0,&RF13MINT_H
5266 3f 40 7a f8        MOV.W       #0xf87a,addr
526a 7e 40 93 00        MOV.B       #0x93,R14

                        zeroize trend record table and history table: we zeroize 0x93...
                        LAB_526e                                      XREF[1]:     5276(j)
526e 8f 43 00 00        MOV.W       #0,0x0(addr)=>trend_index
5272 2f 53              INCD.W      addr
5274 7e 53              ADD.B       #-1,len
5276 fb 23              JNE         LAB_526e
5278 e2 b2 c3 1c        BIT.B       #4,&DAT_1cc3
527c 16 28              JNC         LAB_52aa
527e 3f 40 66 f8        MOV.W       #0xf866,addr
5282 7e 40 09 00        MOV.B       #0x9,len

                        zeroize 0x09 words after the expiration indicator in the head...
                        LAB_5286                                      XREF[1]:     528e(j)
5286 8f 43 00 00        MOV.W       #0,0x0(addr)=>DAT_f866
528a 2f 53              INCD.W      addr
528c 7e 53              ADD.B       #-1,len
528e fb 23              JNE         LAB_5286
5290 c2 43 65 f8        MOV.B       #0,&fram_expirationindicator                  Start a new life!
5294 0f 43              MOV.W       #0,addr
```

We (nearly) know how to reset a sensor

- Set Stage of Life byte
- Set Activity Switch byte
- Clean up the Glucose records section: this also resets the *wear time count*

But we need to **compute correct CRCs** for section we patch!

Computing a CRC shouldn't be difficult, right?

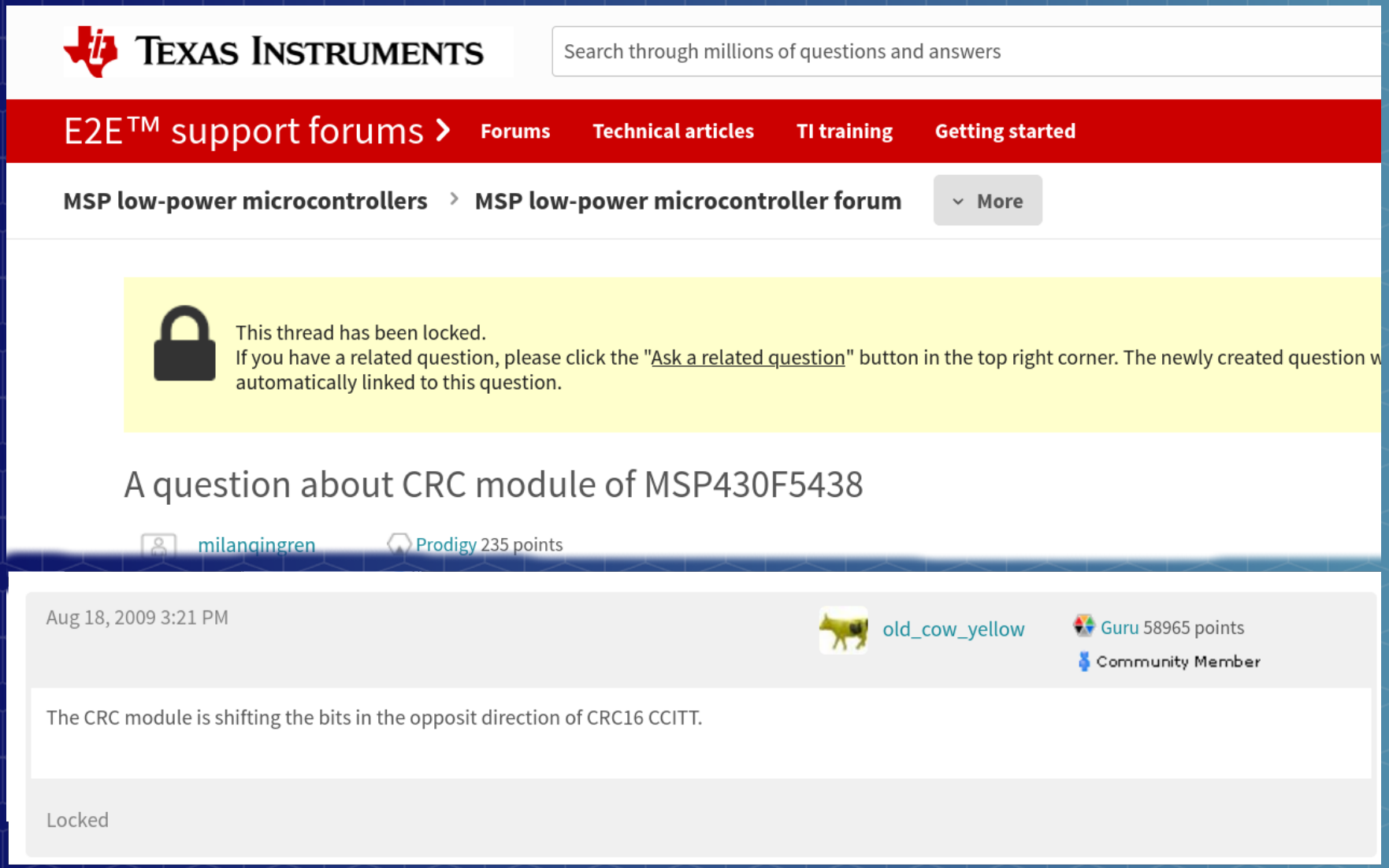
Which one is it? ...

Algorithm	Result	Check	Poly	Init	RefIn	RefOut	XorOut
CRC-16/CCITT-FALSE	0x29B1	0x29B1	0x1021	0xFFFF	false	false	0x0000
CRC-16/ARC	0xBB3D	0xBB3D	0x8005	0x0000	true	true	0x0000
CRC-16/AUG-CCITT	0xE5CC	0xE5CC	0x1021	0x1D0F	false	false	0x0000
CRC-16/BUYPASS	0xFEE8	0xFEE8	0x8005	0x0000	false	false	0x0000
CRC-16/CDMA2000	0x4C06	0x4C06	0xC867	0xFFFF	false	false	0x0000
CRC-16/DDS-110	0x9ECF	0x9ECF	0x8005	0x800D	false	false	0x0000
CRC-16/DECT-R	0x007E	0x007E	0x0589	0x0000	false	false	0x0001
CRC-16/DECT-X	0x007F	0x007F	0x0589	0x0000	false	false	0x0000
CRC-16/DNP	0xEA82	0xEA82	0x3D65	0x0000	true	true	0xFFFF
CRC-16/EN-13757	0xC2B7	0xC2B7	0x3D65	0x0000	false	false	0xFFFF
CRC-16/GENIBUS	0xD64E	0xD64E	0x1021	0xFFFF	false	false	0xFFFF
CRC-16/MAXIM	0x44C2	0x44C2	0x8005	0x0000	true	true	0xFFFF
CRC-16/MCRF4XX	0x6F91	0x6F91	0x1021	0xFFFF	true	true	0x0000
CRC-16/RIELLO	0x63D0	0x63D0	0x1021	0xB2AA	true	true	0x0000
CRC-16/T10-DIF	0xD0DB	0xD0DB	0x8BB7	0x0000	false	false	0x0000
CRC-16/TELEDISK	0x0FB3	0x0FB3	0xA097	0x0000	false	false	0x0000
CRC-16/TMS37157	0x26B1	0x26B1	0x1021	0x89EC	true	true	0x0000
CRC-16/USB	0xB4C8	0xB4C8	0x8005	0xFFFF	true	true	0xFFFF
CRC-A	0xBF05	0xBF05	0x1021	0xC6C6	true	true	0x0000
CRC-16/KERMIT	0x2189	0x2189	0x1021	0x0000	true	true	0x0000
CRC-16/MODBUS	0x4B37	0x4B37	0x8005	0xFFFF	true	true	0x0000
CRC-16/X-25	0x906E	0x906E	0x1021	0xFFFF	true	true	0xFFFF

Tried them all, none matched!

To be honest, several months past before we found the solution...

Solution

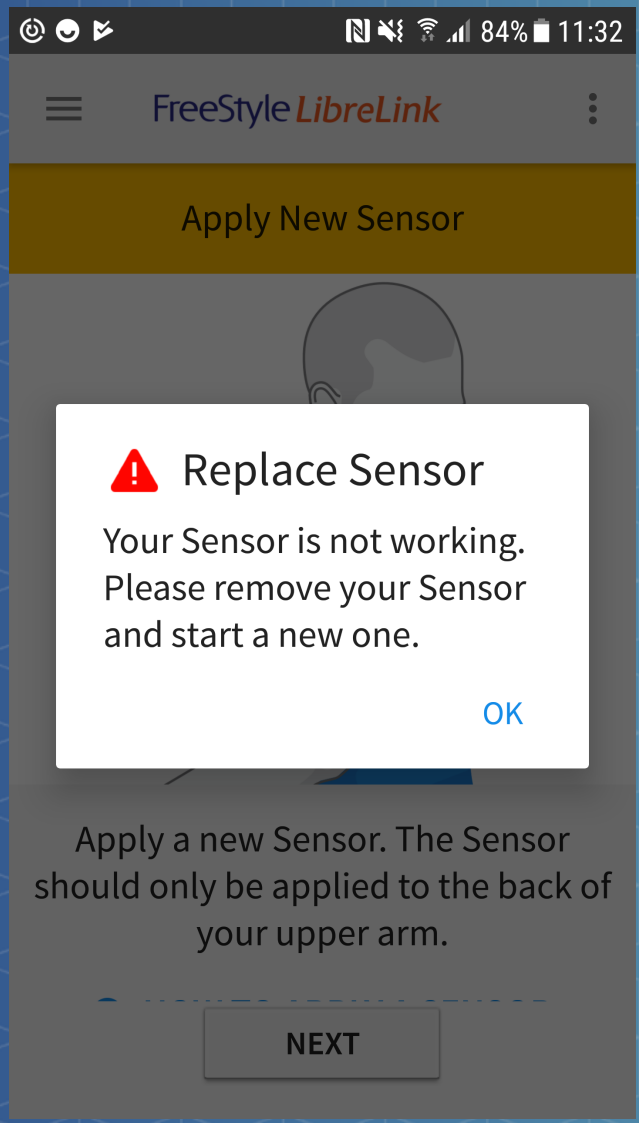


Shifts bits in the opposite direction

Kill a sensor

We know how to *resurrect* a sensor. An attacker may want to do the opposite: **kill** a sensor.

- Corrupt the memory of the sensor. Quick, easy and dirty.
- Or set Stage of Life to 5 (or 6).



Corrupt memory

Wanna hack? Working around limitationss

1. Max life time:**HACKED**
2. **Warm up time**
3. Geographical location

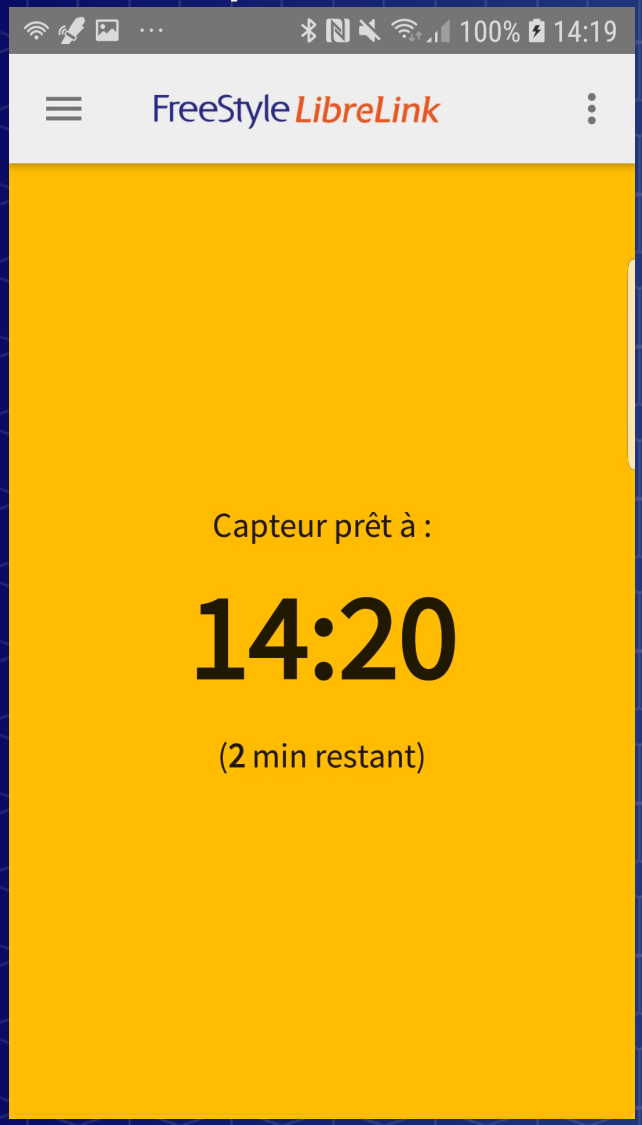
Demo: Set up

Sensor

[*] Hack PatchTimeValues: we set warmup=5 weartime=6912000 minutes

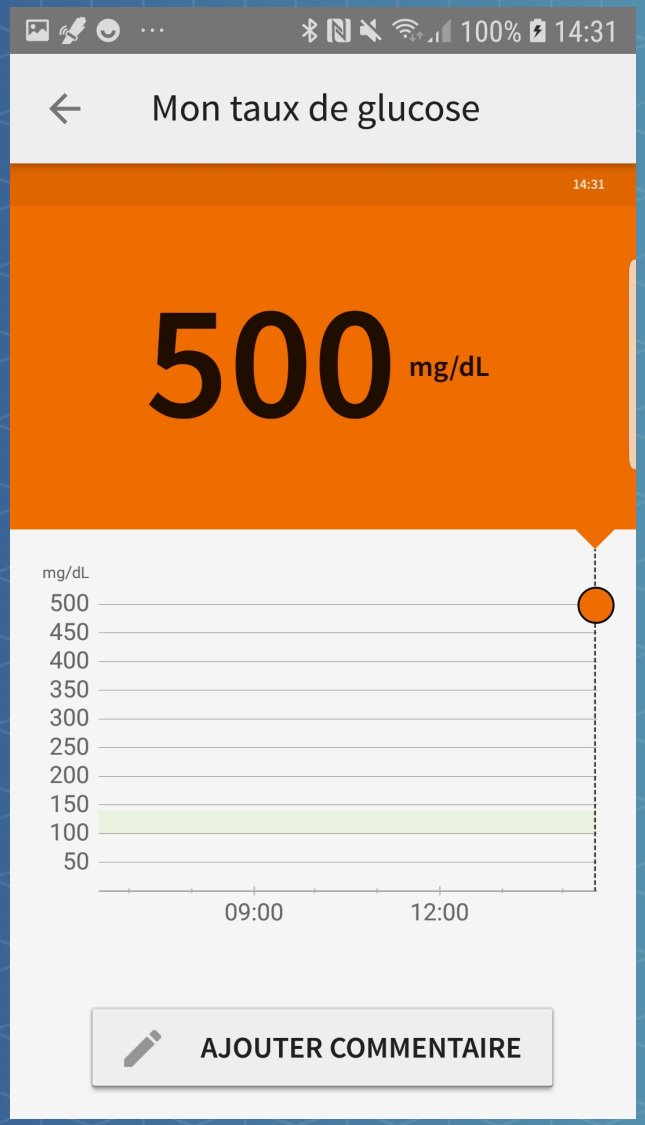
Show time

Backup slides ;P



Warm up time modified to 2 minutes

Wear limit hacked to 4800 days



We can hack glucose value with a Frida hook

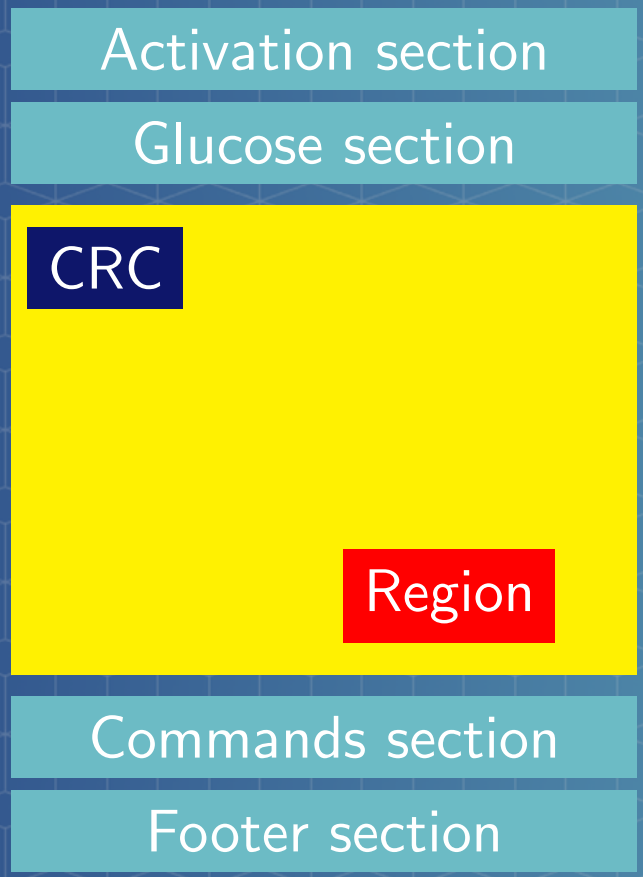
Wanna hack? Working around limitations

1. Max life time:**HACKED**
2. Warm up time: **HACKED**
3. **Geographical location**

Sensor region

- Sensor region is located in the *sensor section*
- Flip region indicator
- Recompute CRC of section
- Activate sensor

Code	Geographic region
01	Europe/UK
02	US 10-day sensors
08	Israel



Close up on the sensor section in memory

Wanna hack? Working around limitations

1. Max life time:**HACKED**
2. Warm up time: **HACKED**
3. Geographical location: **HACKED**

Requires NFC proximity + secret password

Conclusion

We bypass all limitations

although, globally, the design is good / has been done with care

Mitigation

For an attacker, it is **far easier** to:

- Infect the victim's phone with a ransomware
- Or create a fake diabetes app

The weakest link is the smartphone

Debate: can we secure smartphones for critical uses?

References

- Security analysis of a Connected Glucose Sensor, Technical report
- GoodV Android application
- Readdump.py
- NFC exploitation with RF430RFL152 and 'TAL152, PoC || GTFO, 20:03
- Presentation at BlackAlps 2019

Thank You

Contact us:
@cryptax @travisgoodspeed

Thanks to:
Anonymous diabetic contacts :) and
@aamirlakhani @PagetPhil @TuxDePoinsisse @aurelsec
@passthesaltcon